

**Lattice Boltzmann Metodu ve Grafik Kartlarıyla Paralel
Programlama Kullanarak Mikrokanallarda
Tek Fazlı Akış ve Isı Transferi Analizi**

Proje No: 110M750

Yürütücü : Y. Doç. Dr. Cüneyt SERT
Araştırmacı : Y. Doç. Dr. Barbaros ÇETİN
Araştırmacı : Berat ÇELİK

ŞUBAT 2012
ANKARA

ÖNSÖZ

Bir senelik çalışmayı kapsayan bu TÜBİTAK 1002 projesinde mikrokanallarda tek fazlı akış ve ısı transferi benzetimleri yapabilmek amacı ile Lattice Boltzmann Metodu kullanılarak bir kod geliştirilmiştir. Projenin yürütücülüğünü ODTÜ Makina Mühendisliği'nden Y. Doç. Dr. Cüneyt Sert yapmıştır. Bilkent Üniversitesi Makina Mühendisliği Bölümü'nden Y. Doç. Dr. Barbaros Çetin projeye araştırmacı olarak destek vermiştir. ODTÜ Makina Mühendisliği'nde yüksek lisans eğitimine devam eden S. Berat Çelik projede bursiyer olarak görev almıştır.

İÇİNDEKİLER

Tablo Listesi	4
Şekil Listesi	4
Özet	5
Abstract	5
1. Giriş ve Genel Bilgiler	6
2. Gereç ve Yöntem	8
2.1. Lattice Boltzmann Metodu	8
2.1.1. Çarpışma Hesabı	10
2.1.2. Haberleşme (Güncelleme) Hesabı	10
2.1.3. Makroskobik Parametrelerin Hesaplanması	11
2.1.4. Sınır Şartlarının Uygulanması	11
2.1.4.1. Girişte Hız Sınır Şartı	11
2.1.4.2. Sabit Duvarda Hız Sınır Şartı	12
2.1.4.3. Çıkışta Hız Sınır Şartı	12
2.1.4.4. Sıcaklık Sınır Şartı	13
2.1.4.5. Duvarda Kayma Hızı Sınır Şartı	13
2.1.4.6. Duvarda Sıcaklık Atlaması Sınır Şartı	14
2.2. Kanal İçerisinde Hız Profili ve Nusselt Hesabı	15
2.3. LBM Simülasyon Esasları	16
2.4. MATLAB ve Jacket ile LBM'in Grafik Kartlarında Paralel Programlanması	17
3. Bulgular ve Tartışma	17
3.1. CPU-GPU Performansının Karşılaştırılması	22
4. Sonuç	23
Referanslar	24
TÜBİTAK Proje Özet Bilgi Formu	26

TABLO LİSTESİ

Tablo 1. Tam gelişmiş akış için Nusselt sayısının Kn ve κ ile değişimi	21
Tablo 2. Farklı sıklıkta ağlarda Nusselt sayısının Kn ile değişimi	21
Tablo 3. Kanal akışı için GPU ve CPU ile alınan çözüm süreleri	23

ŞEKİL LİSTESİ

Şekil 1. Akış rejimlerinin ve kullanılan denklemlerin Kn sayısına göre sınıflandırılması	6
Şekil 2. D2Q9 modelinde kullanılan 9 hareket yönü	9
Şekil 3. Eşit Δx ve Δy aralıklarına sahip $M \times N$ noktalı Kartezyen ağ	10
Şekil 4. Örnek bir akış ve ağ için birinci dağılım fonksiyonun haberleşme işlemi sırasındaki değişimi	11
Şekil 5. Kanal akışı için çözülen boyutsuz problem ve lattice modeli	17
Şekil 6. $Kn = 0$ için kanal içerisinde farklı konumlardaki hız profillerinin (üst) ve kanal boyunca oluşan boyutsuz basınç düşüşünün (alt) referans sonlu eleman çözümü ile karşılaştırması	18
Şekil 7. Farklı Kn sayıları için kanal içerisinde tam gelişmiş hız dağılımlarının analitik çözüm ile karşılaştırması	19
Şekil 8. $Kn = 0$ için kanal içerisinde farklı konumlardaki ortalama sıcaklık kullanılarak boyutsuzlaştırılmış sıcaklık dağılımları (T/T_{mean}) ve referans sonlu eleman çözümü ile karşılaştırması	19
Şekil 9. (a-c) Üçer farklı κ ve Kn değeri için Nu sayısının kanal boyunca değişimi ve (d) gelişen rejimdeki Nu sayısının referans HAD çözümü (BEJAN, 2004) ile karşılaştırılması	20
Şekil 10. Kanal akışı için CPU ve GPU kullanan kodların çalışma süreleri	23

ÖZET

Bu projede iki boyutlu mikrokanallarda, kaygan akış rejiminde tek fazlı akış ve ısı transferi problemi sayısal olarak çalışılmıştır. Bu amaçla Lattice Boltzmann Metodu (LBM) kullanılarak bir kod geliştirilmiştir. MATLAB ve Jacket yazılımları kullanılarak geliştirilen kodun önemli bir özelliği grafik kartları (GPU) üzerinde paralel çalışabilmesidir. Geliştirilen kod ile mikro ve makro kanallarda benzetimler yapılmış, elde edilen hız profilleri ve Nusselt sayıları literatürdeki Navier-Stokes tabanlı analitik ve sayısal sonuçlarla karşılaştırılmış ve uyumlu sonuçlar alınabildiği gösterilmiştir. Kaygan akış rejimini kapsayan Knudsen sayısı aralığı için yapılan mikrokanal benzetimlerinde hız kayması ve sıcaklık atlaması sınır şartları kullanılmıştır. Geliştirilen kodun grafik kartı üzerindeki paralel performansı ana işlemci üzerinde çalışan hali ile kıyaslanmış ve belli bir büyüklükten sonraki ağlarda 10 kattan daha büyük hızlanmalar elde etmenin mümkün olabildiği gösterilmiştir.

Anahtar Kelimeler : Lattice Boltzmann Metodu (LBM), Mikrokanal, Kaygan akış rejimi, Grafik kartlarında (GPU) paralel programlama, CUDA, Jacket

ABSTRACT

In this project fluid flow and heat transfer in two-dimensional micro channels are studied numerically. A computer code based on Lattice Boltzmann Method (LBM) is developed for this purpose. The code is developed using MATLAB and Jacket software and has the important feature of being able to run parallel on Graphics Processing Units GPU). The code is used to simulate flow and heat transfer inside micro and macro channels. Obtained velocity profiles and Nusselt numbers are compared with the Navier-Stokes based analytical and numerical results available in the literature and good matches are observed. Slip velocity and temperature jump boundary conditions are used for the micro channel simulations with Knudsen number values covering the slip flow regime. Speed of the parallel version of the developed code running on a GPU is compared with that of the serial running CPU version and for large enough meshes more than 10 times speed up is reported.

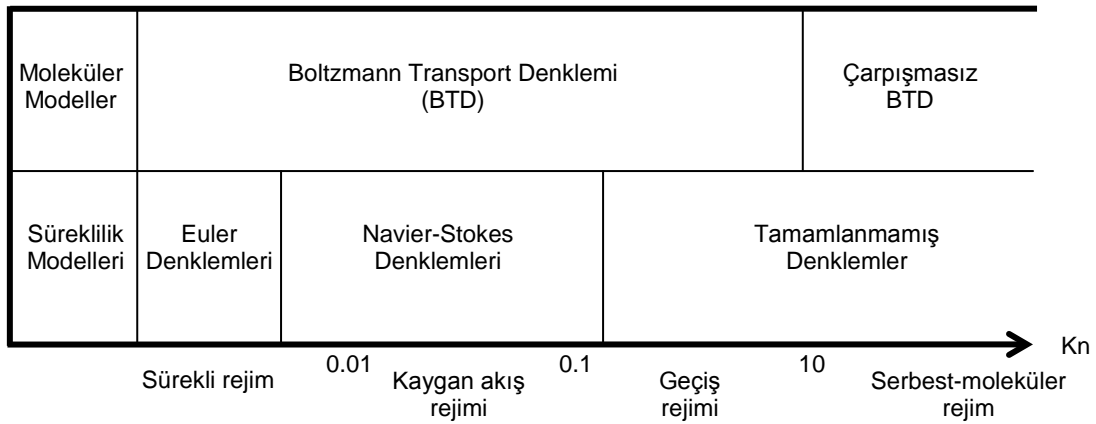
Keywords : Lattice Boltzmann Method (LBM), Micro channel, Slip flow regime, Parallel computing on Graphics Processing Units (GPU), CUDA, Jacket

1. GİRİŞ ve GENEL BİLGİLER

Elektronik parçalardaki küçülme, mikro fabrikasyon ve nano teknolojideki gelişmelerle birlikte ilerlemektedir. Bu küçülme tıp alanında (hücrelerin ya da bakterilerin ayrıştırılması-sayılması, DNA üretilmesi vb.), elektronik devrelerin soğutulmasında, uzay endüstrisinde, Mikro Elektro Mekanik Sistemlerde (MEMS) ve benzeri teknolojilerde ilerlemeye neden olmuştur. Bu sistemlerin pek çoğu akış ve ısı transferi içerir ve genellikle 1 mikrondan büyük 1000 mikrondan küçük kanallar ihtiva ederler. Bu cihazların optimum tasarım parametrelerinin belirlenebilmesi için mikrokannallar içerisindeki akış ve ısı transferinin hassas bir biçimde modellenenebilmesi ve sayısal benzetiminin yapılabilmesi önemlidir.

Akış problemlerini modellemek için temel olarak iki yöntem vardır. Birincisi, akışkanın birbirlerinden ayrı, ama etkileşim halinde olan çok sayıda molekülden meydana geldiğini kabul eden moleküler model, diğeri ise akışkanın sürekli ve devamlı bölünebilir olduğunu kabul eden süreklilik (continuum) modelidir. Makro ölçekte, genel olarak akışkan sürekli kabul edilir, yani akışkanın hızı, yoğunluğu, basıncı, vb. özelliklerinin uzaydaki her noktada, her an için sürekli fonksiyonlarla tanımlanabildiği varsayılır. Süreklilik modelinde, Navier-Stokes denklemleri olarak bilinen kütle, momentum ve enerji korunumu denklemlerinin çözümü ile hız, basınç, sıcaklık, vb. dağılımlar elde edilebilir. Ama süreklilik kabulünün geçersiz olduğu mikro ve nano akışlarda bu denklemlerin kullanılması uygun olmamaktadır.

Çözülme istenen problemin karakteristik boyutu (L) akışkan moleküllerinin birbirlerine çarpmadan aldığı ortalama mesafeye (mean free path λ), yaklaştıkça süreklilik varsayımı geçerliliğini yitirmektedir. Bu iki uzunluğunun birbirine oranı Knudsen (Kn) sayısı olarak bilinen boyutsuz bir niceliktir. Kn sayısı büyüdükçe akışkan seyrelmekte ve akışın modellenmesi sürekli modelden moleküler modele doğru kaymaktadır. Şekil 1’de görüldüğü gibi küçük Kn sayıları için akış sürekli, büyük Kn sayıları için serbest-moleküler kabul edilir. $0.01 < Kn < 0.1$ değerleri için akış kaygan-akış rejiminde, $0.1 < Kn < 10$ değerleri için de geçiş rejimindedir.



Şekil 1. Akış rejimlerinin ve kullanılan denklemlerin Kn sayısına göre sınıflandırılması

Sürekli rejimdeki bir problem için Navier-Stokes denklemlerinin çözülmesini temel alan klasik Hesaplamalı Akışkanlar Dinamiği (HAD) yöntemleri kullanılabilir. Fakat akış geçiş ve serbest moleküler rejime doğru kaydığında Navier-Stokes denklemleri geçerliliğini kaybetmekte, Boltzmann Transport Denklemi (BTD) problemin fizikini daha doğru modellemeye başlamaktadır. Ama BTD 7 boyutlu (3 uzaysal, 3 hız ve 1 zaman boyutu), doğrusal olmayan, integro-differansiyel bir denklemdir ve bugün bilinen yöntemlerle çözülmesi uygulanabilir değildir. Sürekli ve kaygan akış rejimlerinde Navier-Stokes tabanlı HAD teknikleri, geçiş rejiminde de istatistiksel tabanlı Monte Carlo yöntemi

yaygın olarak kullanılmaktadır. Serbest-moleküler rejimde ise çözümler çarpışmasız BTM kullanılarak yapılmaya çalışılmaktadır. Bu çalışmanın konusunu oluşturan ve BTM tabanlı bir yöntem olan Lattice Boltzmann Metodu (LBM) bütün bu rejimleri modelleyebilme potansiyeli olan bir sayısal teknik olarak karşımıza çıkmaktadır.

LBM diğer moleküler metotlarla (Direct Simulation of Monte Carlo, Molecular Dynamics) karşılaştırıldığında çok belirgin ayırt edici özellikler sunar. Bunlar

- iki fazlı akışlar ve gözenekli akışlar gibi karmaşık problemlere kolay uygulanabilmesi,
- daha hızlı çalışması,
- daha az hafıza kullanması,
- paralel programlamaya yatkın olması ve
- kolay programlanabilir olmasıdır.

Bu avantajları sayesinde popülerliği artmakta olan LBM halen üzerinde aktif olarak çalışılan ve geliştirilmeye açık bir metottur.

Sayısal metotlar kullanarak çözülmek istenen problemlerin fiziksel karmaşıklığı ve elde edilmek istenen sayısal hassasiyet her geçen gün artmakta ve tek bir işlemciyi kullanan seri programlar bu zahmetli hesaplamalar için yetersiz kalmaktadır. Günümüzde HAD programları rutin olarak paylaşımlı ve paylaşımsız hafıza mimarisine sahip bilgisayarlarda, OpenMP ve MPI benzeri yazılım kütüphaneleri yardımı ile paralel olarak çalıştırılmaktadır. Ancak son yıllarda grafik kartlarının (GPU) performanslarının ana işlemci (CPU) performanslarına göre çok daha hızlı artması ve bu kartların genel amaçlı hesaplama için kullanılabilirliğini hedefleyen programlama modellerinin geliştirilmesi ile birlikte GPU'lar yüksek başarılı bilimsel hesaplama ihtiyaçları için tercih edilir hale gelmiştir. Son yıllarda GPU tabanlı çalışan akış çözücülerinin seri akış çözücülerine kıyasla onlarca kata varan mertebelerde hızlandırılabilirdiği ortaya konulmuştur.

GPU'ları yüksek başarılı hesaplama yapmaya uygun kılan donanım özelliklerinin başında çok sayıda çekirdekli işlemcilere sahip olmaları gelmektedir. Örneğin NVIDIA firmasının özel olarak yüksek başarılı hesaplama amacı ile ürettiği Tesla kartlarında eş zamanlı işlem yapabilen 960 çekirdek mevcuttur. Bu sayı günümüz standart masaüstü bilgisayarlarında kullanılan CPU'larda 8 ile sınırlıdır. GPU'ların hafızaya erişim özellikleri de CPU'lardan farklıdır. Günümüz hesaplamalarındaki en önemli dar boğazlardan biri olan hafızaya erişimdeki bant genişliği GPU'lar üzerindeki özel bellekler sayesinde CPU'lara kıyasla çok fazladır. Örnek olarak Tesla kartının hafızaya erişim bant genişliği 100 GB/sn'nin üzerindeyken, Intel'in yeni nesil çok çekirdekli CPU'larında bu miktar 10 GB/sn mertebelerindedir.

Bu üstün özellikleri GPU'ları genel amaçlı bilimsel hesaplamalar için kullanılmaya uygun hale getirir de yakın geçmişe kadar bu çok sınırlı olarak yapılmıştır. Bunun temel nedeni GPU'larla etkileşimin OpenGL veya Direct3D gibi grafik işleme kütüphaneleri üzerinden yapılması zorunluluğudur. Bu ise yapılması gereken bilimsel hesaplama işlerinin klasik grafik hesaplama algoritmalarına uyarlanmasını gerektirir ve büyük programlama becerisi ister. Ancak NVIDIA firması 2007 yılında CUDA (Compute Unified Device Architecture) ismini verdiği paralel hesaplama modeli ve beraberinde sunduğu program geliştirme araçları ile GPU'lar üzerinde genel amaçlı hesaplamalar yapmayı kolaylaştırmıştır. CUDA'nın kullanıma sunulması ile birlikte GPU'ların, içlerinde HAD çalışmalarının da olduğu, bilimsel hesaplama araştırmalarında kullanımı yaygınlaşmaya başlamıştır.

Grafik kartlarının bilimsel hesaplama için sunduğu potansiyel anlaşıldıkça yazılım firmaları bu işi kolaylaştırmak için yardımcı yazılım kütüphaneleri geliştirmeye başlamışlardır. Bu projede kullanılan Jacket yazılımı da CUDA ile GPU programlama becerisini MATLAB ortamına taşıyan, doğrusal cebir

problemlerinin çözümüne odaklanan bir kütüphanedir. Bu çalışmada standart bir LBM çözümünün en çok zaman aldığı bilinen kısımları Jacket yazılımı ile paralelleştirilmiştir.

Bu projede Lattice Boltzmann Metodu grafik kartları üzerinde paralel çalışacak şekilde MATLAB ortamında programlanmıştır. Öncelikle makro ölçekte kanal içerisindeki akış problemi LBM ile çözülmüş, sonrasında mikrokandaki akış problemi kaygan akış rejiminde hız kayması ve sıcaklık atlaması sınır şartları kullanılarak çalışılmıştır. Sonuçlar Navier-Stokes tabanlı analitik ve sayısal sonuçlarla karşılaştırılmıştır. Geliştirilen LBM kodunun GPU'larda paralel çalışan versiyonunun seri çalışan CPU versiyonuna göre, yeterince büyük ağlarda, 10 kattan daha fazla hızlanma sunabildiği gösterilmiştir.

2. GEREÇ VE YÖNTEM

2.1. Lattice Boltzmann Metodu (LBM)

LBM'nda çözülen ana denklem aşağıda verilen Boltzmann Transport Denklemi (BTD)'dir.

$$\frac{\partial f}{\partial t} + (\mathbf{c} \cdot \nabla) f = \Omega \quad (1)$$

Bu denklemdeki f pozisyona, moleküler hıza (\mathbf{c}) ve zamana (t) bağlı dağılım fonksiyonudur. f , belli bir t zamanında, diferansiyel bir kontrol hacminde ve diferansiyel bir hız aralığındaki ($\mathbf{c}, \mathbf{c} + d\mathbf{c}$) akışkan parçacığı sayısını temsil eder. Ω parçacıkların çarpışmasını modelleyen ve çift katlı integral içeren bir terimdir. BTD bu hali ile 7 boyutlu (3 uzaysal boyut, 3 hız boyutu ve bir de zaman boyutu), doğrusal olmayan, integro-diferansiyel bir denklemdir ve bugün bilinen yöntemlerle çözülmesi pratik değildir. Bu denklem genellikle Bhatnagar-Gross-Krook (BGK) çarpışma modeli ile aşağıdaki gibi basitleştirilir (BHATNAGAR, 1954).

$$\Omega = \omega (f^{eq} - f) \quad (2)$$

Bu denklemde ω çarpışma frekansı, f^{eq} ise denge dağılım fonksiyonudur. BGK basitleştirmesi kullanılan BTD uzay ve zamanda aşağıdaki gibi ayrıştırılabilir

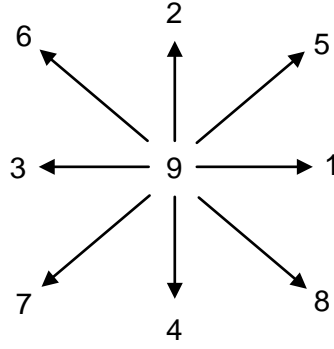
$$f(x + \Delta x, t + \Delta t) = f(x, t)(1 - \omega) + \omega f^{eq}(x, t) \quad (3)$$

ve Chapman-Enskog açılımı kullanılarak çarpışma frekansı akışkanın viskozitesi ile aşağıdaki gibi ilişkilendirilebilir (MOHAMAD, 2011).

$$\omega = \frac{1}{0.5 + \nu \frac{3\Delta t}{(\Delta x)^2}} \quad (4)$$

Bu denklemde ν kinematik viskozite, Δt zaman adımı, Δx ise ağ aralığıdır.

LBM'de ağ noktalarındaki parçacıkların sadece belirli yönlerde hareket etmelerine izin verilir ve bu yönler kullanılan modele göre farklılık gösterir. LBM terminolojinde DXQY olarak gösterilen modellerde X problemin uzay boyutunu Y ise ağ noktalarında kullanılacak hareket yönlerinin sayısını göstermektedir. Akış içermeyen iki-boyutlu problemlerde D2Q4 veya D2Q9 modelleri, akış içeren problemlerde ise D2Q9 modelinin kullanımı yaygındır. Bu modellerin oluşturulmasında Kinetik Teori tabanlı detaylı ve teorik bir türetme prosedürü izlenmektedir (HE, 1997). Bu çalışmada da kullanılan D2Q9 modelindeki hareket yönleri Şekil 2'de gösterilmiştir.



Şekil 2. D2Q9 modelinde kullanılan 9 hareket yönü

Kullanılan sayısal ağın herhangi bir n noktasında, k yönü ile ilişkilendirilen denge dağılım fonksiyonunu aşağıdaki gibi yazmak mümkündür (MOHAMAD, 2011).

$$f_{k,n}^{eq} = w_k \rho_n \left[1 + \frac{\mathbf{c}_k \cdot \mathbf{u}_n}{c_s^2} + \frac{1}{2} \frac{(\mathbf{c}_k \cdot \mathbf{u}_n)^2}{c_s^4} - \frac{1}{2} \frac{|\mathbf{u}_n|^2}{c_s^2} \right] \quad (5)$$

Bu denklemde \mathbf{u}_n ve ρ_n , n noktasındaki makroskobik hız ve makroskobik özkütle değerleridir. $c_s (= 1/\sqrt{3})$ LBM ağının ses hızıdır. \mathbf{c}_k , k yönündeki moleküler hız vektörünü temsil eder. Kullanılan D2Q9 modeli için bu vektörler aşağıdaki gibidir.

$$\mathbf{c}_k = \begin{cases} (1,0) & k = 1 \\ (0,1) & k = 2 \\ (-1,0) & k = 3 \\ (0,-1) & k = 4 \\ (1,1) & k = 5 \\ (-1,1) & k = 6 \\ (-1,-1) & k = 7 \\ (1,-1) & k = 8 \\ (0,0) & k = 9 \end{cases} \quad (6)$$

Denklem (5)'teki w_k değerleri Kinetik Teori kullanılarak elde edilebilen k yönünde hareket eden parçacıkların n noktasındaki tüm parçacıklara oranıdır. D2Q9 modeli için w_k değerleri aşağıdaki gibidir.

$$w_k = \begin{cases} 1/9 & k = 1, 2, 3, 4 \\ 1/36 & k = 5, 6, 7, 8 \\ 4/9 & k = 9 \end{cases} \quad (7)$$

Buraya kadar verilen denklemler kullanılarak detayları ileride anlatılacak olan işlemler sonrasında bir akış alanındaki hız dağılımını hesaplamak mümkündür. Hız hesabının ardından gelen sıcaklık hesabı için daha önce tanımlanan hız dağılım fonksiyonuna benzer şekilde bir ısı dağılım fonksiyonu tanımlanır ve çözülür. Sıcaklık alanı çözümü için ilk olarak Denklem (3)'dekine benzer şekilde ısı dağılım fonksiyonu aşağıdaki gibi ayrıştırılır.

$$g(x + \Delta x, t + \Delta t) = g(x, t)(1 - \omega_t) + \omega_t g^{eq}(x, t) \quad (8)$$

Chapman-Enskog açılımı kullanılarak ısı çarpışma frekansı (ω_t) akışkanın ısıl yayınma katsayısı (α) ile aşağıdaki gibi ilişkilendirilebilir (MOHAMAD, 2011).

$$\omega_t = \frac{1}{0.5 + \alpha \frac{3\Delta t}{(\Delta x)^2}} \quad (9)$$

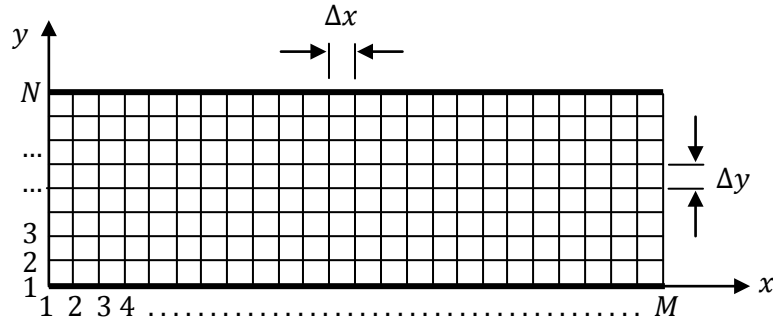
Bu çalışmada sıcaklık hesabı için de hız hesabında olduğu gibi D2Q9 modeli kullanılmıştır.

Denklem (8)'deki ısıl denge dağılım fonksiyonu Denklem (5)'dekine benzer şekilde aşağıdaki gibi yazılabilir (MOHAMAD, 2011).

$$g_{k,n}^{eq} = w_k T_n \left[1 + \frac{c_k \cdot u_n}{c_s^2} \right] \quad (10)$$

Bu denklemde T_n , n noktasındaki makroskobik sıcaklıktır. c_k ve w_k değerleri hız çözümünde kullanılanlarla aynıdır.

LBM'nda çözüm için gereken ağ genellikle Şekil 3'dekine benzer şekilde eşit ve sabit Δx ve Δy değerlerine sahip Kartezyen bir ağ olacak şekilde yaratılır. Bu kısıtlama istenilen bölgelerde ağı istenilen miktarda sıklaştırıp seyrekleştirme konusunda sorun oluştursa da daha karmaşık bir ağa ve formülasyona bu çalışmada gerek duyulmamıştır. Bu çalışmadaki tüm çözümler Δx ve Δy değerleri 1.0 alınarak elde edilmiştir.



Şekil 3. Eşit Δx ve Δy aralıklarına sahip $M \times N$ noktalı Kartezyen ağ

LBM'nda çözüme ağ noktalarında tanımlanmış olan dağılım fonksiyonlarının sıfır olduğu bir ilk şarttan başlanarak zaman döngüsü içinde LBM'nun temelini oluşturan ve aşağıda detayları verilen 4 işlem sırasıyla yapılır.

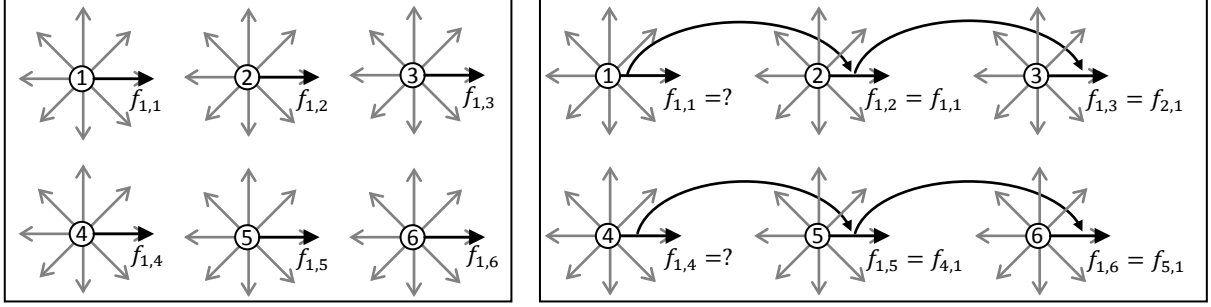
2.1.1. Çarpışma Hesabı

Çarpışma basamağında her ağ noktasındaki yeni dağılım fonksiyonları Denklem (3-7) kullanılarak, ısıl dağılım fonksiyonları da Denklem (8-10) kullanılarak hesaplanır.

2.1.2 Haberleşme (Güncelleme) Hesabı

Haberleşme (güncelleme) işleminde daha önce her bir ağ noktası için ayrı ayrı hesaplanan dağılım fonksiyonları komşu noktalara transfer edilir. Böylece moleküllerin hareketi, yani akış gerçekleşir. Şekil 4'de haberleşmenin nasıl gerçekleştiği 6 noktalı bir örnek ağ üzerinde gösterilmiştir. Her noktadaki 1. dağılım fonksiyonlarının haberleşme öncesinde ve sonrasında aldığı değerler şekilde belirtilmiştir. Benzer şekilde diğer dağılım fonksiyonları da kendi doğrultularında komşu noktalara taşınacaklardır. Bu tek yönlü örnek akışta 1. ve 4. noktaların yer aldığı sol sınır girişi, 3. ve 6. noktalarının bulunduğu sağ sınır ise çıkışı temsil etmektedir. Şekilde de görüldüğü gibi haberleşme

sonrasında $f_{1,1}$ ve $f_{1,4}$ fonksiyonları için komşu noktalardan taşınacak değerler yoktur. Bunlar giriş sınır koşulu kullanılarak hesaplanacaktır.



Şekil 4. Örnek bir akış ve ağ için birinci dağılım fonksiyonunun haberleşme işlemi sırasındaki değişimi

2.1.3. Makroskobik Parametrelerin Hesaplanması

LBM metodunda her bir noktada hesaplanan dağılım fonksiyonlarının toplamı kullanılarak o noktaya ait hidrodinamik basınç ve sıcaklık aşağıdaki gibi hesaplanır

$$p_n = c_s^2 \sum_{k=1}^9 f_{n,k} \quad (11)$$

$$T_n = \sum_{k=1}^9 g_{n,k} \quad (12)$$

Basınç hesabından sonra makroskobik özkütle ve hız aşağıdaki gibi hesaplanabilir

$$\rho_n = \frac{p_n}{c_s^2} \quad (13)$$

$$\mathbf{u}_n = \frac{1}{\rho_n} \sum_{k=1}^9 f_{n,k} \mathbf{c}_{n,k} \quad (14)$$

2.1.4. Sınır Şartlarının Uygulanması

2.1.4.1 Girişte Hız Sınır Şartı

Şekil 3'deki gibi bir kanal akışı için sol sınırda x yönündeki hızın sıfırdan büyük ve sabit, y yönündeki hızın ise sıfır olduğunu düşünelim. Kanal girişindeki bir n noktası için kanalın içine doğru yönelen $f_{1,n}$, $f_{5,n}$ ve $f_{8,n}$ dağılım fonksiyonlarının hesaplanması gerekir. Bunun için girişe dik yöndeki dağılım fonksiyonlarının dengede olmayan bileşenlerinin eşitliği kullanılabilir.

$$f_{1,n}^{non-eq} = f_{3,n}^{non-eq} \quad (15 a)$$

$$f_{k,n}^{non-eq} = f_{k,n} - f_{k,n}^{eq} \quad (15 b)$$

$$f_{1,n} - f_{1,n}^{eq} = f_{3,n} - f_{3,n}^{eq} \quad (16)$$

Bu denklemdeki $f_{1,n}^{eq}$ ve $f_{3,n}^{eq}$ Denklem (5) kullanılarak hesaplanır. Bu hesap için gerekli olan ρ_n , u_n ve v_n değerleri Denklem (13) ve (14)'ü sağlamaktadır. ρ_n değeri bilinmezken u_n girişteki bilinen hıza, v_n ise sifıra eşittir. Tüm bu bilgi ve denklemler bir arada kullanıldığında girişteki ρ_n değeri ile $f_{1,n}$, $f_{5,n}$ ve $f_{8,n}$ dağılım fonksiyonları aşağıdaki gibi hesaplanabilir.

$$\rho_n = \frac{[f_{2,n} + f_{4,n} + f_{9,n} + 2(f_{3,n} + f_{6,n} + f_{7,n})]}{1 - u_n} \quad (17)$$

$$f_{1,n} = f_{3,n} + \frac{2}{3}\rho_n u_n \quad (18)$$

$$f_{5,n} = f_{7,n} + \frac{1}{2}(f_{4,n} - f_{2,n}) + \frac{\rho_n u_n}{6} \quad (19)$$

$$f_{8,n} = f_{6,n} - \frac{1}{2}(f_{4,n} - f_{2,n}) + \frac{\rho_n u_n}{6} \quad (20)$$

2.1.4.2. Sabit Duvar Hız Sınır Şartı

Kayma hızının hesaba katılmadığı durumlarda duvar üzerindeki akışkan hızı duvarın hızına eşit kabul edilir. Şekil 3'deki örnek kanalın alt ve üst duvarlarının hareketsiz olduğunu kabul edersek giriş sınır şartı için yukarıda açıklanan prosedüre benzer şekilde alt duvar için aşağıdaki dağılım fonksiyonları hesaplanabilir.

$$f_{2,n} = f_{4,n} \quad (21)$$

$$f_{5,n} = f_{7,n} - \frac{1}{2}(f_{1,n} - f_{3,n}) \quad (22)$$

$$f_{6,n} = f_{8,n} + \frac{1}{2}(f_{1,n} - f_{3,n}) \quad (23)$$

Benzer denklemler üst duvar için de kullanılabilir.

2.1.4.3. Çıkışta Hız Sınır Şartı

Çıkış sınırında akışın tam olarak gelişmiş olduğu ve hız değerlerinin akış yönünde değişmedikleri kabul edilir. Şekil 3'deki örnek kanal akışında sağ taraftaki sınır çıkış olarak kabul edilirse bu sınırda gerekli olan ρ_n , $f_{3,n}$, $f_{6,n}$ ve $f_{7,n}$ değerleri aşağıdaki gibi hesaplanabilir.

$$\rho_n = \frac{[f_{2,n} + f_{4,n} + f_{9,n} + 2(f_{1,n} + f_{5,n} + f_{8,n})]}{1 + u_n} \quad (24)$$

$$f_{3,n} = f_{1,n} - \frac{2}{3}\rho_n u_n \quad (25)$$

$$f_{7,n} = f_{5,n} - \frac{1}{2}(f_{4,n} - f_{2,n}) - \frac{\rho_n u_n}{6} \quad (26)$$

$$f_{6,n} = f_{8,n} + \frac{1}{2}(f_{4,n} - f_{2,n}) - \frac{\rho_n u_n}{6} \quad (27)$$

y yönüne dik bir çıkış sınırında v_n sıfır alınırken u_n çıkış sınırına yakın bölgede değişmediği kabul edilen hız değeridir. Böylelikle yeterince uzun bir kanalda çıkış sınırına yakın bölgede hızın değişmemesi şartı sağlanmış olur.

2.1.4.4. Sıcaklık Sınır Şartları

Duvar üzerindeki sıcaklık sınır şartı akı korunumu ilkesi ile hesaplanabilir (MOHAMAD, 2011). Bilinmeyen ısıl dağılım fonksiyonları alt duvar için g_2 , g_5 ve g_6 , üst duvar için g_4 , g_7 ve g_8 'dir.

Alt Duvar:

$$g_{2,n}^{eq} - g_{2,n} + g_{4,n}^{eq} - g_{4,n} = 0 \quad (28)$$

$$g_{5,n}^{eq} - g_5 + g_{7,n}^{eq} - g_{7,n} = 0 \quad (29)$$

$$g_{6,n}^{eq} - g_{6,n} + g_{8,n}^{eq} - g_{8,n} = 0 \quad (30)$$

Denklem (10)'daki denge dağılım fonksiyonları Denklem (28-30)'da yerlerine yazıldığında alt duvar için gereken ısıl dağılım fonksiyonları aşağıdaki gibi bulunur.

$$g_{2,n} = T_w(w(2) + w(4)) - g_{4,n} \quad (31)$$

$$g_{5,n} = T_w(w(5) + w(7)) - g_{7,n} \quad (32)$$

$$g_{6,n} = T_w(w(6) + w(8)) - g_{8,n} \quad (33)$$

Üst duvardaki sıcaklık sınır şartı da benzer şekilde çözüme dahil edilebilir.

Çıkış sıcaklık sınır şartı, dışa değer bulma yöntemi ile çıkış sınırında bilinmeyen g_3 , g_6 ve g_7 ısıl dağılım fonksiyonlarının hesaplanmasıyla sağlanır.

$$g_{3,n} = 2g_{3,n-1} - g_{3,n-2} \quad (34)$$

$$g_{6,n} = 2g_{6,n-1} - g_{6,n-2} \quad (35)$$

$$g_{7,n} = 2g_{7,n-1} - g_{7,n-2} \quad (36)$$

Denklem (34-36)'da n çıkış sınırı üzerindeki, $n - 1$ çıkıştan bir önceki, $n - 2$ çıkıştan iki önceki noktaları temsil eder.

2.1.4.5. Duvarda Kayma Hızı Sınır Şartı

Makro kanallarda duvar üzerinde akış hızı sıfır olarak kabul edilirken mikrokanallarda duvar üzerinde hızın sıfır olmadığı ve bu hızın Knudsen sayısının bir fonksiyonu olduğu gösterilmiştir (GAD-EL-HAK, 2004). Şekil 3'deki kanalın alt ve üst duvarlarındaki bir n noktası için kayma hızları aşağıdaki gibi ifade edilebilir (TIAN, 2007).

$$\text{Alt duvar : } u_n^s = u_n - u_n^w = \lambda \left(\frac{\partial u}{\partial y} \right)_n \quad (37)$$

$$\text{Üst duvar : } u_n^s = u_n^w - u_n = \lambda \left(\frac{\partial u}{\partial y} \right)_n \quad (38)$$

Sabit bir duvar için duvar hızı (u_n^w) sıfıra eşitlendiğinde duvarlardaki akışkanın hızları ile kayma hızları (u_n^s) birbirine eşit olur ve bu değerler Denklem (37-38)'deki birinci türevler 1. dereceden geriye fark yaklaşımı ile basitleştirildiğinde aşağıdaki gibi hesaplanabilir.

$$\text{Alt duvar : } u_n = \frac{\lambda(4u_{n+1} - u_{n+2})}{2\Delta y + 3\lambda} \quad (39)$$

$$\text{Üst duvar : } u_n = \frac{\lambda(4u_{n-1} - u_{n-2})}{2\Delta y + 3\lambda} \quad (40)$$

Şekil 3'deki gibi y yönünde N ağ noktası bulunan bir kanal için $\lambda = Kn (N - 1)$ ve $\Delta y = 1$ 'dir. u_{n+1} ve u_{n+2} hızları alt duvardaki bir noktanın hemen üstündeki iki hızı, u_{n-1} ve u_{n-2} ise üst duvardaki bir noktanın hemen altındaki iki hızı temsil eder. Duvarlardaki hız değerleri Denklem (39-40) ile elde edildikten sonra duvarlarda gerekli özkütle ve dağılım fonksiyonları aşağıdaki gibi hesaplanır.

Alt duvar:

$$\rho_n = f_{1,n} + f_{3,n} + f_{9,n} + 2(f_{4,n} + f_{7,n} + f_{8,n}) \quad (41)$$

$$f_{2,n} = f_{4,n} \quad (42)$$

$$f_{5,n} = \frac{\rho_n(1 + u_n)}{2} - (f_{1,n} + f_{8,n}) - \frac{f_{2,n} + f_{4,n} + f_{9,n}}{2} \quad (43)$$

$$f_{6,n} = \frac{\rho_n(1 - u_n)}{2} - (f_{3,n} + f_{7,n}) - \frac{f_{2,n} + f_{4,n} + f_{9,n}}{2} \quad (44)$$

Benzer bir türetme üst duvar için de yapılabilir.

2.1.4.6. Sıcaklık Atlaması Sınır Şartı

Kayma hızı şartındakine benzer şekilde mikrokanallarda duvar üzerindeki sıcaklık duvara en yakın yerdeki akışın sıcaklığından farklı olabilir. Sıcaklık atlaması olarak adlandırılan bu durum matematiksel olarak aşağıdaki gibi ifade edilebilir (TIAN, 2007).

$$\text{Alt duvar : } T_n^s = T_n - T_n^w = \alpha \left(\frac{2\gamma}{\gamma + 1} \right) \left(\frac{\lambda}{Pr} \right) \left(\frac{\partial T}{\partial y} \right)_n \quad (45)$$

$$\text{Üst duvar : } T_n^s = T_n^w - T_n = \alpha \left(\frac{2\gamma}{\gamma + 1} \right) \left(\frac{\lambda}{Pr} \right) \left(\frac{\partial T}{\partial y} \right)_n \quad (46)$$

Bu denklemde T_n^w bilinen duvar sıcaklığı, T_n^s duvardaki sıcaklık atlaması, T_n duvarın komşuluğundaki akışkanın sıcaklığı, Pr Prandtl sayısıdır. Denklemleri basitleştirmek için aşağıdaki sıcaklık atlaması katsayısı tanımından faydalanılabilir.

$$C = \lambda\kappa = \lambda \alpha \left(\frac{2\gamma}{\gamma + 1} \right) \left(\frac{1}{Pr} \right) \quad (47)$$

Denklem (45-46)'daki sıcaklık türevleri 1. dereceden geriye fark tekniği ile ifade edildiğinde gerekli olan duvar üzerindeki akışkan sıcaklığı hesaplanır.

$$\text{Alt duvar : } T_n = \frac{C((4T_{n+1} - T_{n+2}) + 2\Delta y T_n^w)}{2\Delta y + 3C_j} \quad (48)$$

$$\text{Üst duvar : } T_n = \frac{C((4T_{n-1} - T_{n-2}) + 2\Delta y T_n^w)}{2\Delta y + 3C_j} \quad (49)$$

Bulunan bu sıcaklık değerleri alt ve üst duvarlarda aşağıdaki dağılım fonksiyonlarını hesaplamak için kullanılır. Denklem (31-33)'deki T değerleri yerine Denklem (48) ve (49)'da hesaplanan T değerleri konulduğunda aşağıdaki sıcaklık atlama sınırı şartı elde edilir.

Alt duvar:

$$g_{2,n} = \frac{2}{9}T_n - g_{4,n} \quad , \quad g_{5,n} = \frac{T_n}{18} - g_{7,n} \quad , \quad g_{6,n} = \frac{T_n}{18} - g_{8,n} \quad (50)$$

Benzer bir türetme üst duvar için de yapılabilir.

2.2. Kanal İçerisindeki Hız Profili ve Nusselt Hesabı

Makro akışlarda paralel duvarlar arasındaki hız profili Navier-Stokes denklemlerinin çözümünden elde edilir.

$$\mu \frac{\partial^2 u}{\partial y^2} = \frac{\Delta P}{L} \quad (51)$$

Alt ve üst duvarlardaki sınır koşulları aşağıdaki gibidir

$$u_0 = Kn \frac{\partial u}{\partial y} \quad (52)$$

$$u_H = -Kn \frac{\partial u}{\partial y}$$

İlgili diferansiyel denklemin çözümü parabolik bir eğridir. Kanalın birimsiz yüksekliği $\eta = y/H$ olarak tanımlandığında analitik hız profili aşağıdaki gibi elde edilir.

$$\frac{u}{u_{ort}} = -\frac{6(\eta^2 - \eta - Kn)}{6Kn + 1} \quad (53)$$

Duvar ve akışkan arasında ısı transferinin karakteristiği Nusselt sayısı ile belirlenebilir. Yeterince uzun bir kanalda hız ve sıcaklık profillerinin dengeye ulaştığı anda Nusselt sayısının sabitlendiği görülür. Duvar yüzeyinde kondüksiyon ve konveksiyon ile gerçekleşen ısı transferlerinin toplamı sıfır olacaktır.

$$h(T_w - T_{mean}) - k \left(\frac{\partial T}{\partial y} \right) = 0 \quad (54)$$

Aşağıdaki iki birimsiz değişken tanımlanır

$$\eta = \frac{y}{H} \quad (55)$$

$$\theta = \frac{T_{mean} - T_w}{T_i - T_w} \quad (56)$$

ve ortalama sıcaklık aşağıdaki gibi ifade edilirse

$$T_{mean} = \frac{\int_A \rho u T \partial A}{\int_A \rho u \partial A} \quad (57)$$

Denklem (54) aşağıdaki hali alır

$$\frac{k}{H} (T_i - T_w) \left(\frac{\partial \theta}{\partial \eta} \right) = h (T_w - T_{mean}) \quad (58)$$

$$\frac{\partial \theta}{\partial \eta} = \frac{2H}{k} \frac{h (T_w - T_{mean})}{T_i - T_w} \frac{1}{2} \quad (59)$$

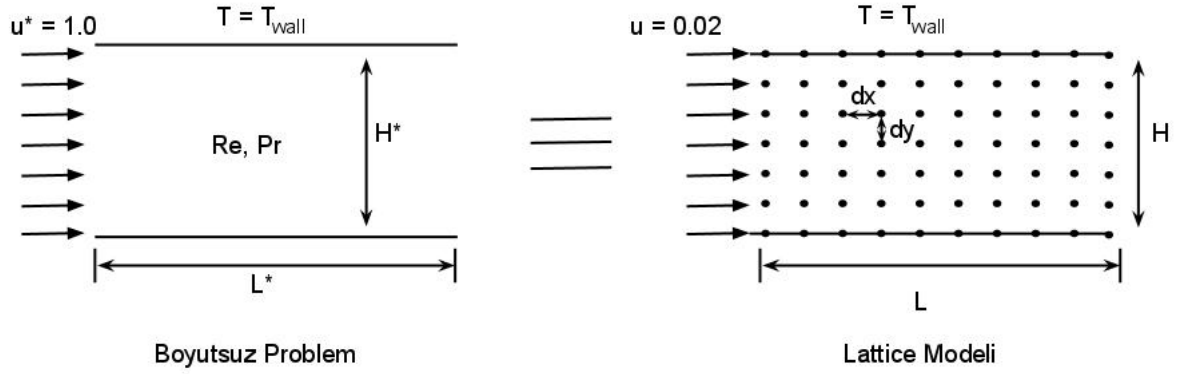
$Nu = 2Hh/k$ olarak tanımlanan Nusselt sayısı birimsiz sıcaklık cinsinden aşağıdaki gibi yazılabilir

$$Nu = - \frac{2}{\theta} \frac{\partial \theta}{\partial \eta} \quad (60)$$

2.3. LBM Simülasyonu Esasları

Bir LBM simülasyonu yapılırken öncelikle çözülmek istenen problemin lattice modeli oluşturulmalıdır. Bu projede çalışılan kanal akışı için çözülmek istenen boyutsuz problem ve buna karşılık gelen lattice modeli Şekil 5’de verilmiştir. Lattice modelinde $\Delta x = \Delta y = 1.0$ olarak alınmıştır. İki problem arasında geometrik benzerliği sağlamak için iki problemin kanal yüksekliğinin kanal boyuna oranı (H/L) ve dinamik benzerliği sağlanmak için de iki problemin Re ve Pr sayıları eşitlenmiştir. Boyutsuz eşitlikler gözetilerek lattice modelinin viskozite ve ısıl yayılım katsayıları hesaplanır. Lattice modelindeki giriş hızı literatürde de tarif edildiği gibi (MOHAMAD, 2011) çözümün stabilitesini sağlamak için $u = 0.02$ alınmıştır. Simülasyon sırasında izlenen basamaklar şöyle sıralanabilir:

1. Boyutsuz problem için örnek olarak $H^* = 1.0$, $L^*/H^* = 20$, $Re = 10$ ve $Pr = 10$ seçilir.
2. Lattice modelinin x -yönündeki (M) ve y -yönündeki ($N = M(H^*/L^*)$) ağ noktası sayısı seçilir ve $H (= N)$ ve $L (= M)$ hesaplanır.
3. Lattice modelin viskozitesi $\nu = uH/Re$ eşitliğinden ve ısıl yayılım katsayısı da $\alpha = Pr \nu$ eşitliğinden bulunur.
4. Lattice modeli için çarpışma frekansı ve ısıl çarpışma frekansı Denklem (4) ve (9) kullanılarak hesaplanır.
5. Kanal içerisindeki dağılım fonksiyonları başlangıçta sıfır olarak atanır.
6. 2.1.1, 2.1.2, 2.1.3 ve 2.1.4 kısımlarında anlatılan işlemler her zaman adımı için tekrarlanır.
7. Zamandan bağımsız çözüme ulaşıncaya kadar simülasyon sonlandırılır.



Şekil 5. Kanal akışı için çözülen boyutsuz problem ve lattice modeli

2.4. MATLAB ve Jacket ile LBM'in Grafik Kartlarında Paralel Programlanması

LBM metodunun paralelleştirmeye uygun olmasının temelinde bir ağ noktası için hesaplanan çarpışma işlemlerinin diğer ağ noktalarında yapılan benzer işlemlerden ve bir dağılım fonksiyonuna uygulanan haberleşme işleminin diğer dağılım fonksiyonlarına uygulananlardan bağımsız olması özelliği yatar. Böylece çarpışma ve haberleşme hesapları GPU üzerindeki çok sayıdaki işlemcide farklı ağ noktaları için aynı anda paralel olarak yapılabilir ve oldukça yüksek paralel hesaplama performansları elde edilebilir. Bu çalışmada kullanım kolaylığı göz önüne alınarak MATLAB ve Jacket yazılımları ile GPU üzerinde paralelleştirme yapılmıştır. Bu iki yazılım kullanıldığında CPU üzerinde çalışan bir kodu GPU'ya uyarlamak için oldukça az miktarda değişiklik gerekmektedir. MATLAB ve Jacket değil de C gibi derlenen bir programlama dili kullanarak daha fazla paralel hızlanma gözlemleyen araştırmacılar olsa da (TUBBS, 2011) çabuk ve zahmetsiz kod geliştirme avantajı düşünüldüğünde MATLAB ve Jacket seçiminin yerinde olduğu söylenebilir.

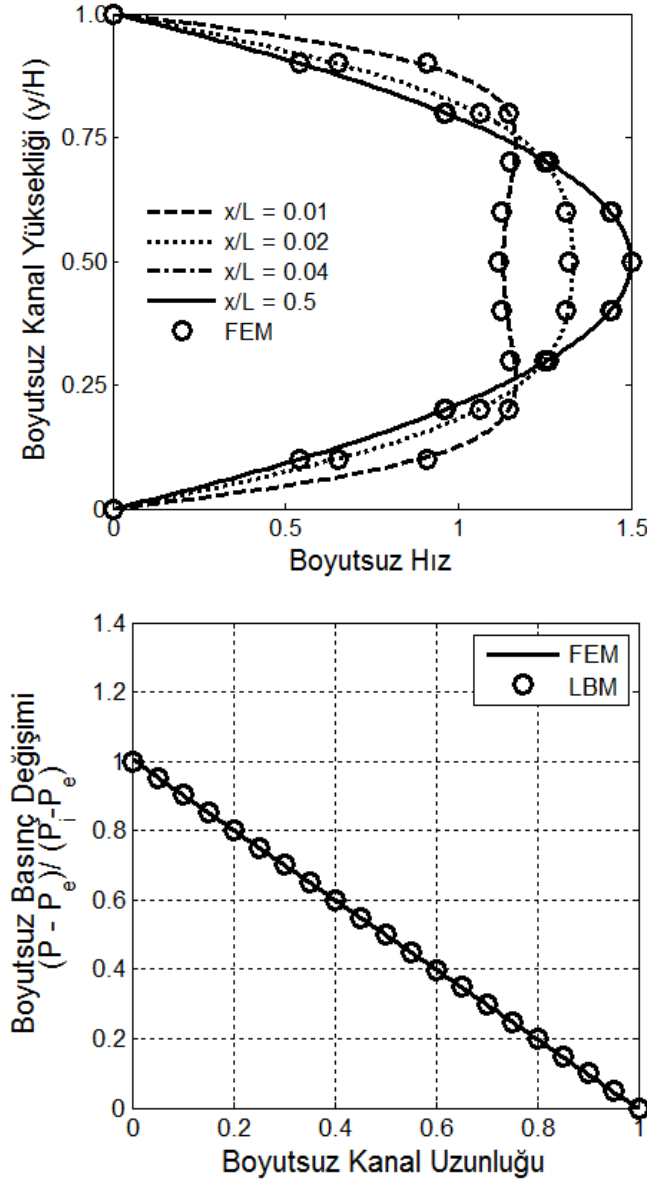
Geliştirilen LBM programının CPU'da ve GPU'da çalışan halleri Ek 1 ve Ek 2'de sunulmuştur. İki kod arasındaki tek önemli fark paralel hesaplamalarda kullanılacak değişkenlerin CPU veya GPU'da tanımlanmış olmalarıdır. Jacket yazılımı GPU hafızasında tanımlanmış değişkenler için otomatik olarak mümkün olan paralel hesaplamaları gerçekleştirir. Böylece CPU üzerinde çalışacak şekilde geliştirilmiş bir MATLAB kodunun GPU'ya transferi çok kolaylaşmaktadır.

3. BULGULAR VE TARTIŞMA

Bu çalışmada mikrokannallar içerisindeki hız ve sıcaklık dağılımları LBM ile hesaplanmıştır. Çözümler sürekli rejimde ve duvarda kayma hızı ve sıcaklık atlaması sınır şartlarının geçerli olduğu kaygan-akış rejiminde ($0 < Kn < 0.1$) gerçekleştirilmiştir. Tüm çözümlerde Re ve Pr sayıları 10 olarak alınmıştır. Sıcak atlaması sınır şartı için Denklem (47)'de tanımlanan κ parametresinin farklı değerleri (0, 1.667, 10) için sonuçlar alınmıştır. $\kappa = 0$, duvarda kayma hızının olduğu ancak sıcaklık atlamasının olmadığı hayali, ancak kayma hızının ısı transferine etkisini gösteren önemli bir durumu temsil etmektedir. $\kappa = 1.667$ hava için tipik bir değer, $\kappa = 10$ ise çok yüksek sıcaklık atlamasını temsil eden bir değerdir (ÇETİN, 2005).

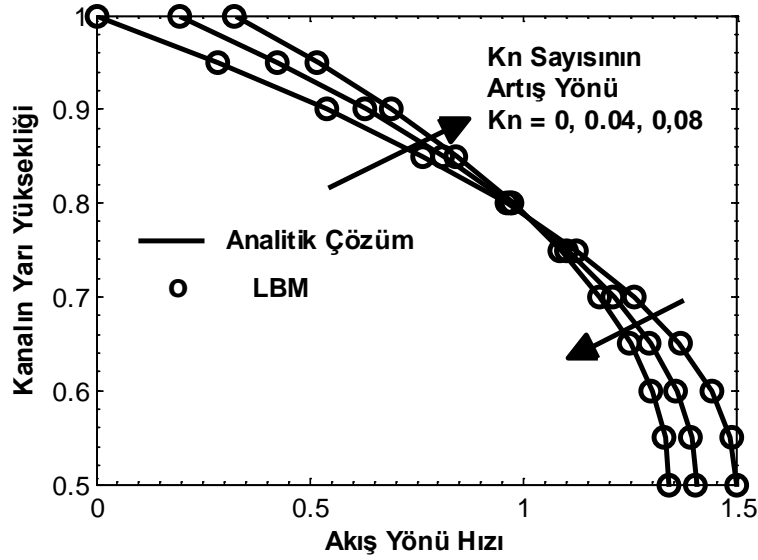
Şekil 6'da bir makro kanal içerisinde farklı konumlardaki hız profilleri $Kn = 0$ için gösterilmiş ve sonlu eleman tabanlı COMSOL yazılımı ile elde edilen çözüm ile karşılaştırma yapılmıştır. Kanal girişinde sabit olarak verilen hız dağılımı kanal içinde çok kısa bir mesafede gelişmekte ve $x/L = 0.04$ 'den sonra tam gelişmiş hız dağılımını yakalanmaktadır. Ayrıca Şekil 6'da aynı makro kanal akışı için kanal boyunca oluşan boyutsuz basınç düşüşü de verilmiş ve LBM sonucunun referans sonlu eleman metodu sonucu ile uyumluluğu gösterilmiştir. Basınç şekilde görüldüğü gibi giriş ve çıkış

basınçları kullanılarak boyutsuzlandırılmıştır. Akışın hidrodinamik olarak geliştiği bölgenin çok kısa olmasından dolayı neredeyse tüm kanal boyunca sabit basınç düşüşü görülmektedir.



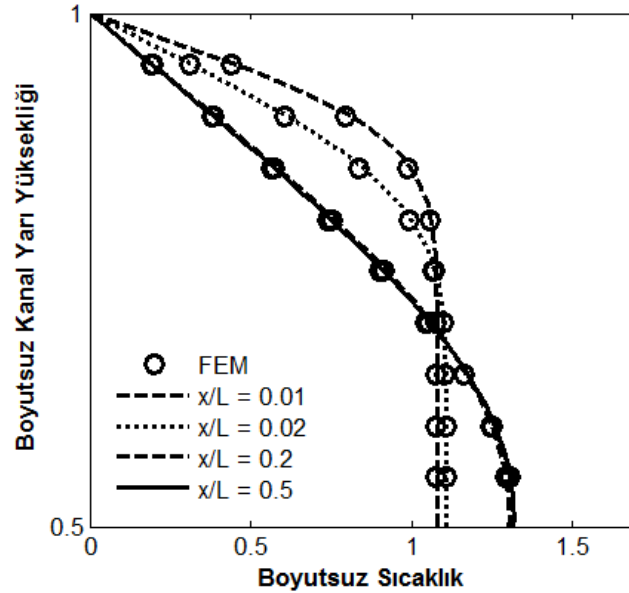
Şekil 6. $Kn = 0$ için kanal içerisinde farklı konumlardaki hız profillerinin (üst) ve kanal boyunca oluşan boyutsuz basınç düşüşünün (alt) referans sonlu eleman çözümü ile karşılaştırması

Şekil 7'de kanal içerisindeki tam gelişmiş hız dağılımları farklı Kn sayıları için gösterilmiş ve analitik çözümler ile karşılaştırma yapılmıştır. Beklenildiği gibi artan Kn sayısı ile duvardaki kayma hızının değeri artmaktadır. LBM sonucu ile analitik çözüm mükemmel bir şekilde örtüşmektedir.



Şekil 7. Farklı Kn sayıları için kanal içerisinde tam gelişmiş hız dağılımlarının analitik çözüm ile karşılaştırması

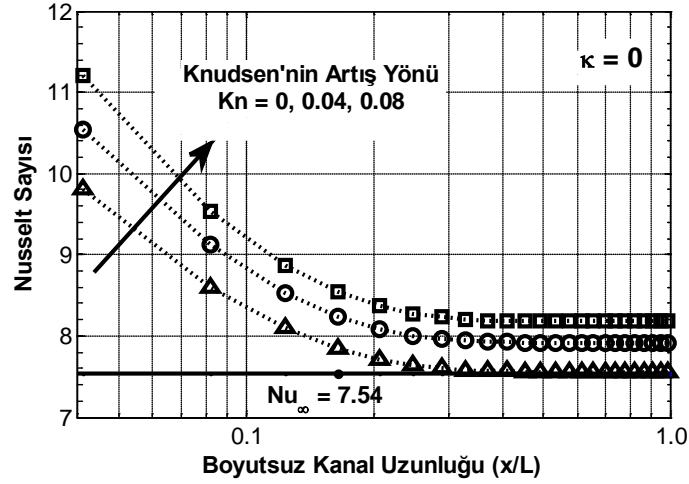
Şekil 8'de kanalın üst yarısında, 4 farklı x konumunda, kesitlerdeki ortalama sıcaklık (Denklem (56) kullanılarak değil) kullanılarak boyutsuzlandırılmış sıcaklık dağılımları $Kn = 0$ için gösterilmiştir. Boyutsuz sıcaklıklar kanal girişinde 1, duvarlarda ise 0 olarak alınmıştır. Beklenildiği gibi duvarda verilen sıfır değeri kanal boyunca yakalanmaktadır ve boyutsuz sıcaklık profili $x/L = 0.2$ 'den sonra değişmemektedir.



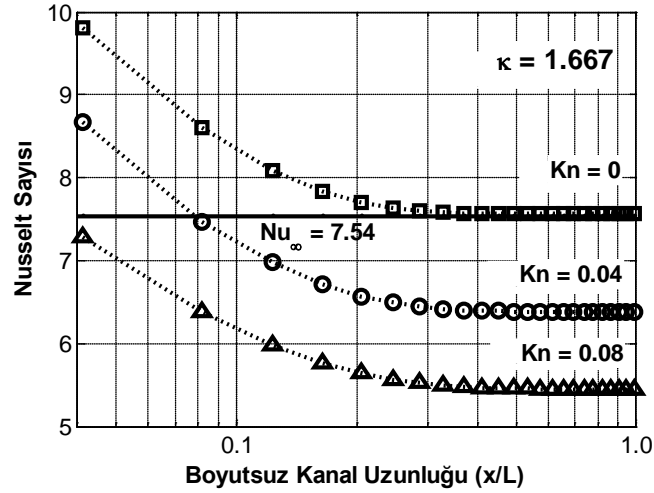
Şekil 8. $Kn = 0$ için kanal içerisinde farklı konumlardaki ortalama sıcaklık kullanılarak boyutsuzlaştırılmış sıcaklık dağılımları (T/T_{mean}) ve referans sonlu eleman çözümü ile karşılaştırması

Şekil 9(a-c)'de Nusselt sayısının kanal boyunca değişimi farklı κ değerleri için gösterilmektedir. Grafiklerde makro kanal için tam gelişmiş Nu sayısı olan 7.54 (INCROPERA, 1996) değeri de gösterilmiştir. LBM sonuçları makro kanal çözümüyle uyum göstermektedir. Tam gelişmiş Nu sayıları daha kapsamlı bir şekilde farklı Kn ve κ değerleri için Tablo 1'de verilmiştir. Şekil 9(d)'de

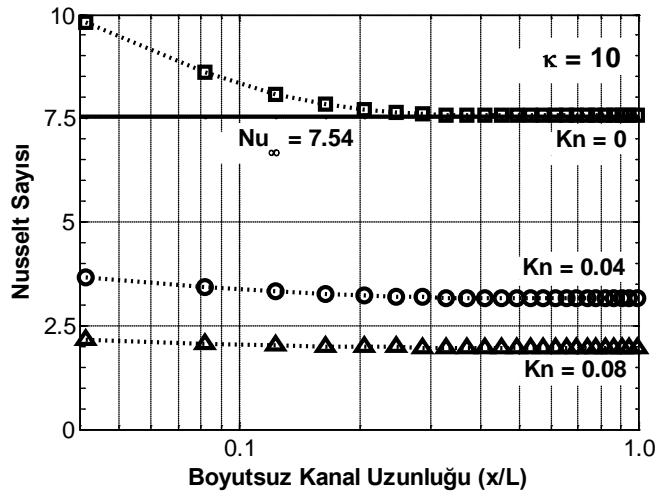
$Kn = 0$ değeri için gelişmekte olan bölgedeki Nu sayısı değişimi literatürdeki referans analitik çözüm (BEJAN, 2004) ile karşılaştırılmış ve farkların % 3'ün altında kaldığı görülmüştür.



(a)

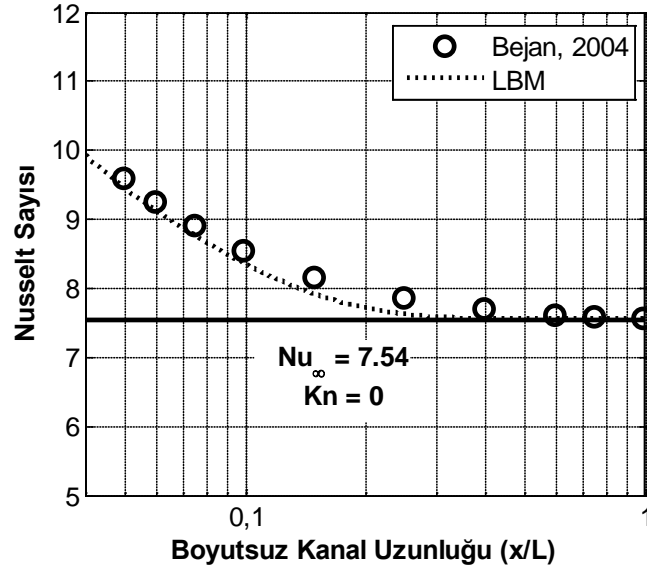


(b)



(c)

Şekil 9. (a-c) Üçer farklı κ ve Kn değeri için Nu sayısının kanal boyunca değişimi ve (d) gelişen rejimdeki Nu sayısının referans HAD çözümü (BEJAN, 2004) ile karşılaştırılması



(d)

Şekil 9 (Devam) (a-c) Üçer farklı κ ve Kn değeri için Nu sayısının kanal boyunca değişimi ve (d) gelişen rejimdeki Nu sayısının referans HAD çözümü (BEJAN, 2004) ile karşılaştırılması

Tablo 1. Tam gelişmiş akış için Nusselt sayısının Kn ve κ ile değişimi

	Kn					
κ	0.00	0.02	0.04	0.06	0.08	0.10
0	7,55	7,75	7,91	8,05	8,18	8,28
1.667	7,55	6,93	6,38	5,88	5,44	5,06
10	7,55	4,48	3,14	2,41	1,95	1,64

Tablo 2'de farklı ağ yoğunluklarında $\kappa=1.667$ için elde edilmiş tam gelişmiş Nu sayıları görülmektedir. Tabloda ayrıca (ÇETİN, 2005) tarafından sonlu farklı yöntemi kullanılarak elde edilmiş Nu sayıları da verilmiştir.

Tablo 2. Farklı sıklıkta ağlarda Nusselt sayısının Kn ile değişimi

	$\kappa = 1.667$					
	$Kn = 0.00$	$Kn = 0.02$	$Kn = 0.04$	$Kn = 0.06$	$Kn = 0.08$	$Kn = 0.10$
21x420	7,577	6,944	6,380	5,881	5,439	5,049
41x820	7,563	6,940	6,383	5,888	5,449	5,060
61x1220	7,559	6,939	6,383	5,888	5,449	5,061
81x1620	7,557	6,937	6,382	5,888	5,449	5,061
ÇETİN, 2005	7,541	6,926	6,374	5,882	5,445	5,053

3.1. CPU-GPU Performansının Karşılaştırılması

Jacket kullanılarak GPU'ya uyarlanan kodun paralel performansını görebilmek için kanal içindeki akış farklı yoğunluktaki ağlarla çözülmüş ve CPU ve GPU üzerinde çalışan kodların çözüm süreleri ölçülmüştür. CPU olarak "Intel Xeon E5620 Quad-Core 2.40 GHz" işlemcinin tek bir çekirdeği, GPU olarak ise "NVIDIA Tesla C1060, 4 GB RAM" kartı kullanılmıştır. Hem CPU'da hem de GPU'da tüm işlemler tek hassasiyetli sayılar kullanılarak yapılmıştır. Bunun sebebi kullanılan grafik kartının çift hassasiyetli sayılar için yeterli performansı verememesidir. Yeni nesil Fermi mimarisindeki kartlarda bu dezavantaj ortadan kalkmıştır. Ancak tek hassasiyet kullanımının elde edilen sonuçların doğruluğuna önemli bir olumsuz etkisi olmadığı da kontrol edilmiştir.

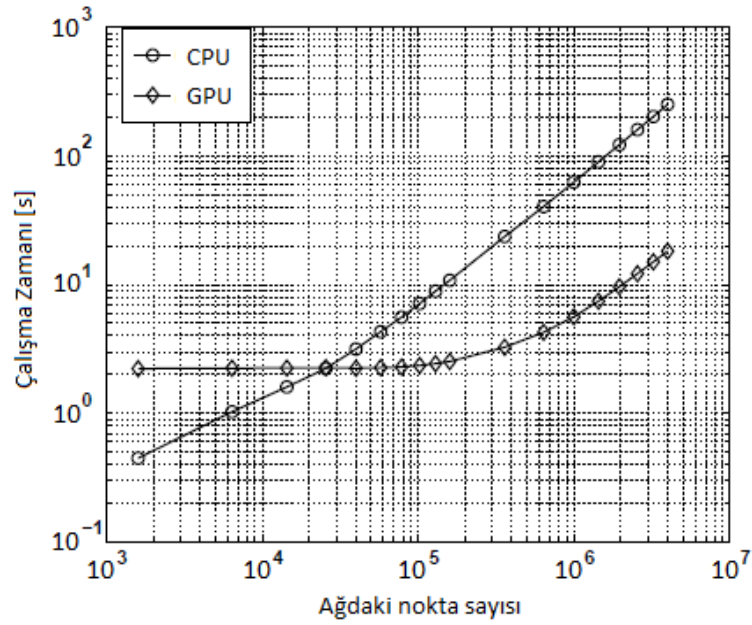
Geliştirilen LBM kodunun CPU ve GPU üzerindeki çalışma süresi ve hızlanma miktarları Tablo 3 ve Şekil 10'da sunulmuştur. Hızlanma değerleri CPU'da harcanan zamanın GPU'da harcanana oranıdır. Sunulan değerler bir makro kanal benzetimine ait olmakla birlikte CPU-GPU hız karşılaştırması κ , Kn , Re gibi akış parametrelerinden hiç etkilenmemektedir. Şekil 10'de görüldüğü gibi toplam 18 farklı ağ kullanılmış, fakat sadelik adına seçilmiş 10 ağa karşılık gelen sonuçlar Tablo 3'e konulmuştur. Tablo 3'de saniye olarak verilen çalışma zamanı değerleri yakınsamış sonuçların elde edilmesi için gereken süreler olmayıp doğru bir şekilde CPU-GPU karşılaştırması yapmaya imkan verecek az miktarda zaman adımı için koşturulmuş kodların çalışma süreleridir. CPU-GPU hız karşılaştırması kodların kaç zaman adımı için çalıştırıldığından bağımsızdır.

Şekil 10'da görüldüğü gibi GPU kullanan kodun çalışma süresi belli büyüklükteki bir ağa kadar sabit kalmaktadır. Buradan çıkarılacak sonuç küçük problemler için GPU hafızasına veri göndermek ve GPU'da hesaplanan verileri CPU'ya geri almak için gereken zamanın diğer işlem zamanlarına baskın olduğudur. Diğer bir deyişle küçük ağlarda GPU'nun tüm kapasitesini kullanarak verimli bir şekilde çalışabileceği kadar çok miktarda hesaplamaya gerek duyulmamaktadır. Bu küçük problemler için GPU bir hızlanma değil, aksine yavaşlama getirmektedir. Yaklaşık 26,000 gibi bir ağ noktasından sonra GPU sürelerinin CPU sürelerinin altına düştüğü gözlenmiştir. Bu noktadan sonra ağ büyüdükçe CPU süreleri hızla artmakta olup denenen en büyük ağda 14 kata varan hızlanma görülmüştür.

Bu sonuçlar göstermektedir ki GPU'nun avantajı büyük problemler için net olarak ortaya çıkmaktadır. Ancak ortalama en fazla 10,000'ler mertebesinde ağ noktasına ihtiyaç duyan iki boyutlu problemlerde GPU kullanımı bir avantaj sağlamamaktadır. Bu gözlemden sonra, iki boyutlu olarak geliştirilen LBM çözücüsü, bu proje çalışması kapsamında olmadığı halde, üç boyuta geçirilmeye başlanmış fakat bu raporda sunulabilecek bir sonuç henüz elde edilememiştir. Şekil 10'da verilen GPU hızlanma miktarları tek bir GPU kullanılarak elde edilmiş değerlerdir. Satın alınan Jacket lisansı bilgisayardaki birden fazla GPU ile çalışma özelliğine sahiptir. Geliştirilen kodun birden fazla GPU kullanacak hale getirilmesi ile birlikte küçük ağlar için de kayda değer hızlanmalar elde edilebileceği öngörülmektedir.

Tablo 3. Kanal akışı için GPU ve CPU ile alınan çözüm süreleri

Ağ boyutu		GPU zamanı (s)	CPU zamanı (s)	Hızlanma
20 x 80 =	1,600	2.22	0.45	0.20
40 x 160 =	6,400	2.24	1.03	0.46
80 x 320 =	25,600	2.26	2.23	0.99
120 x 480 =	57,600	2.26	4.31	1.90
160 x 640 =	102,400	2.35	7.13	3.03
200 x 800 =	160,000	2.53	10.8	4.28
400 x 1600 =	640,000	4.26	40.4	9.49
600 x 2400 =	1,440,000	7.44	90.0	12.1
800 x 3200 =	2,560,000	12.2	160.0	13.2
1000 x 4000 =	4,000,000	18.2	251.0	13.8



Şekil10. Kanal akışı için CPU ve GPU kullanan kodların çalışma süreleri

4. SONUÇ

Bu projede hem makro kanallar hem de mikrokanallar içerisindeki akış ve ısı transferi LBM ile modellenmiştir. Geliştirilen LBM kodu hem CPU üzerinde hem de ekran kartları üzerinde paralel olarak çalıştırılmıştır. Çalışmadan çıkan sonuçlar şöyle özetlenebilir:

- Geliştirilen LBM kodu makro kanal akışlarında ve kaygan akış rejimindeki mikrokanal akışlarında akış ve ısı transferini başarıyla modellemiştir. Alınan sonuçlar literatürdeki analitik ve sayısal sonuçlarla karşılaştırılmış ve güzel bir uyum görülmüştür.

- LBM'in paralelleştirilmeye uygun yapısı sebebiyle GPU üzerinde yapılan denemelerde 10 kattan fazla hızlanma görülmüştür. Ancak bu tip hızlanmalar için üç-boyutlu problemlerin çalışılması gerektiği, iki-boyutlu problemlerin GPU'ların performanslarını tam olarak sergileyebilmeleri için yeterince zorlayıcı olmadıkları anlaşılmıştır. MATLAB ve Jacket ikilisinin GPU'larda bilimsel hesaplama yapmak için çok faydalı olduğu ve kod geliştirme zamanını çok azalttığı görülmüştür.
- Proje kapsamında olmadığı için bu raporda bahsedilmeyen kanal akışı haricinde akışlar da çalışılmış ve geliştirilen LBM kodunun yüksek Reynolds sayılarındaki sıkıştırılmayan akış çözümlerinde çok başarılı performans sergilediği gözlenmiştir. Bu tip çözümlerde LBM'in klasik HAD metotlarında gerek duyulan stabilizasyon teknikleri veya akış yönünü dikkate alan formülasyonlar gibi detaylara gerek duymadığı görülmüştür.
- Ek 1 ve Ek 2'de verilen LBM kodları çok kısa ve anlaşılır bir yapıya sahiptir. Bunun sebebi LBM tekniğinin çok basit bir algoritmaya sahip olmasıdır. Bu kısalık ve basitlik sebebiyle diğer araştırmacılar ve öğrenciler tarafından anlaşılması ve kullanılması da kolaydır. Bu anlamda diğer klasik HAD metotları ile geliştirilen uzun ve karmaşık kodlara göre daha üstündür.

Bu projenin devamında, geliştirilen LBM kodunun geçiş rejimi (transition regime) tabir edilen ve Navier-Stokes tabanlı tekniklerin kullanılamayacağı kanal akışları için de kullanılması planlanmaktadır. Literatürde LBM ile geçiş rejiminde kanal akışını modelleyen çalışmalar bulunmaktadır (SHOKOUHMAND, 2011). Klasik yaklaşım bu rejimdeki bir akışı DSMC metodu ile çözebilmektedir. LBM ise hem sürekli rejimi, hem kaygan-akış rejimini hem de geçiş rejimini modelleyebilen çok güçlü bir yöntemdir ve esas avantajı bunu yapabildiğinde ortaya çıkacaktır. Proje devamında yapılabilecek diğer çalışmalar iki-boyuttan üç-boyuta geçiş ve tek GPU yerine birden fazla GPU kullanacak şekilde kodun geliştirilmesidir.

REFERANSLAR

BHATNAGAR, P. L. , Gross, E. P. and Krook M., A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems, *Phys. Rev.*, 94, 511-525, (1954).

ÇETİN, B., Analysis of Single Phase Convective Heat Transfer in Microtubes and Microchannels, Y. Lisans Tezi, Makina Muh. Bolumu, Orta Doğu Teknik Üniversitesi, (2005).

GAD-EL-HAK, M., Transport phenomena in microdevices, *Journal of Applied Mathematics and Mechanics*, 84, No. 7, 494-498, (2004).

HE, X., Luo, L. Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, *Phys. Rev. E* 56, 6, 6811-6817 (1997)

INCROPERA, F. P., DeWITT, D. P., Fundamentals of Heat and Mass Transfer, John Wiley & Sons, 4. baskı, (1996)

MOHAMAD, A. A., Lattice Boltzmann Method Fundamentals and Engineering Applications with Computer Codes, Springer-Verlag, London (2011). Pp: 45, 47, 54, 63, 70

SHOKOUHMAND, H., Meghdadi I. A. H., An improved thermal lattice Boltzmann model for rarefied

gas flows in wide range of Knudsen number, International Communications in Heat and Mass Transnsfer, 38, 1463-1469, (2011).

TIAN, Z. W., Zou C., Liu H., Gou Z., Liu Z., Zheng C., Lattice Boltzmann scheme for simulating thermal micro-flow, Physica A 385, 59-68, (2007).

TUBBS, K.R., TSAI, F.T.-C., GPU Accelerated Lattice Boltzmann Model for Shallow Water Flow and Mass Transport, Int. J. Numer. Meth in Engrg, 86, pp. 316-334, 2011.

BEJAN, A. Convective Heat Transfer 3rd Edition, Wiley & Sons. Inc., New Jersey (2004), Pp: 135.

TÜBİTAK
PROJE ÖZET BİLGİ FORMU

Proje No: 110M750
Proje Başlığı: Lattice Boltzmann Metodu ve Grafik Kartlarıyla Paralel Programlama Kullanarak Mikrokanallarda Tek Fazlı Akış ve Isı Transferi Analizi
Proje Yürütücüsü ve Araştırmacılar: Y. Doç. Dr. Cüneyt SERT, Y. Doç.Dr. Barbaros ÇETİN, Berat ÇELİK
Projenin Yürütüldüğü Kuruluş ve Adresi: ODTÜ Makina Mühendisliği Bölümü, 06800 Ankara
Destekleyen Kuruluş(ların) Adı ve Adresi: ---
Projenin Başlangıç ve Bitiş Tarihleri: 15/02/2011 – 08/02/2012
Öz (en çok 70 kelime) Bu projede LBM ile iki-boyutlu mikrokanallarda tek fazlı akış ve ısı transferinin analizleri yapılmıştır. Çalışmalar Navier-Stokes denklemlerinin de geçerli olduğu kaygan-akış rejimini kapsamaktadır. Kayma hızı, sıcaklık atlaması ve ısı transferi karakteristikleri incelenerek, LBM ile elde edilen sonuçlar literatürdeki sonuçlarla karşılaştırılmıştır. Geliştirilen LBM tabanlı çözücü yüksek başarımlı hesaplama yapabilmek için özel olarak geliştirilmiş grafik kartları (GPU) üzerinde paralel çalışacak şekilde yazılmış ve CPU-GPU hız karşılaştırmaları yapılmıştır.
Anahtar Kelimeler: Lattice Boltzmann Metodu, Mikrokanal, Kaygan akış rejimi, GPU hesaplama, Paralel programlama, CUDA, Jacket
Fikri Ürün Bildirim Formu Sunuldu mu? Evet <input type="checkbox"/> Gerekli Değil <input checked="" type="checkbox"/> <small>Fikri Ürün Bildirim Formu'nun tesliminden sonra 3 ay içerisinde patent başvurusu yapılmalıdır.</small>
Projeden Yapılan Yayınlar: S. B. Çelik, C. Sert, B. Çetin, "Simulation of Lid-driven Cavity Flow by Parallel Implementation of Lattice Boltzmann Method on GPUs", 2nd Int. Symp. on Computing in Science & Engineering, İzmir, June 2011. S. B. Çelik, C. Sert, B. Çetin, "Simulation of Channel Flow by Parallel Implementation of Thermal Lattice Boltzmann Method on GPU", 7th Int. Conf. on Computational Heat and Mass Transfer, İstanbul, July 2011.